B.TECH

TEC-503

FIFTH SEMESTER EXAMINATION, 2007-2008

DESIGN AND ANALYSIS OF ALGORITHM

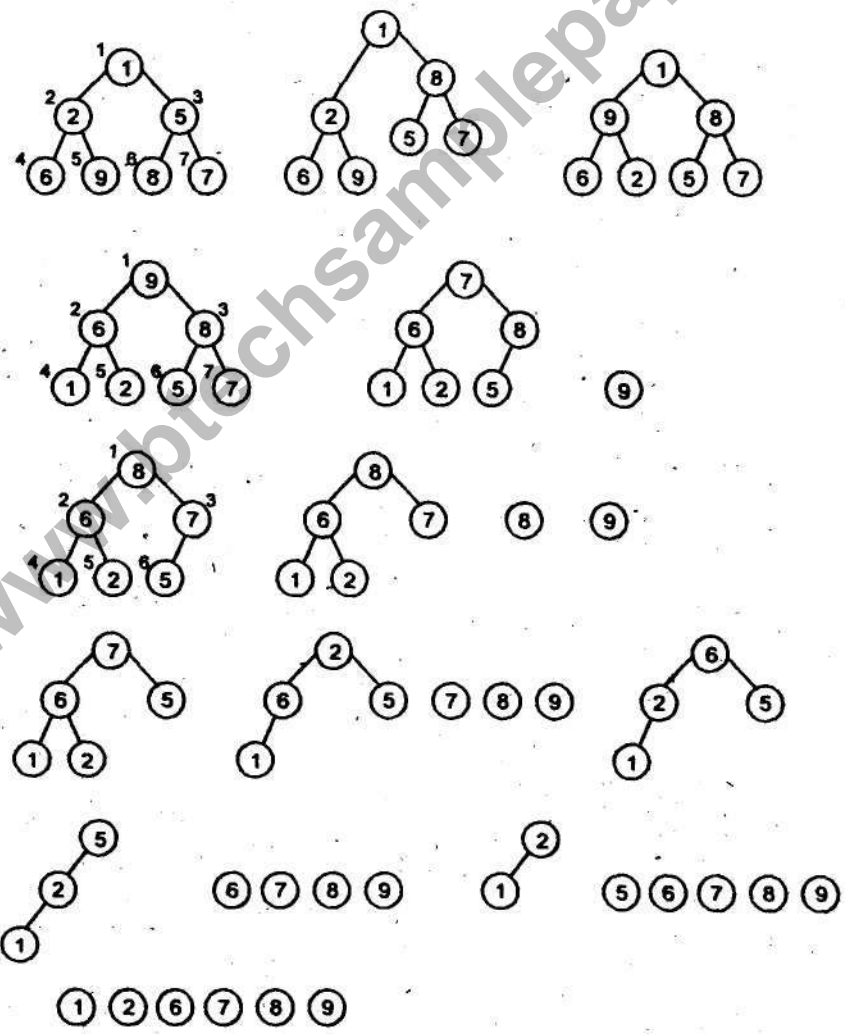
Time: 3 *Hours* *Total Marks:* 100

Note: *Attempt ALL questions.*

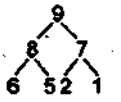
Q.1. Attempt *any four* parts of the following **(5×4=20)**

(*a*) Show the steps in heap sort to arrange following data in non-decreasing order

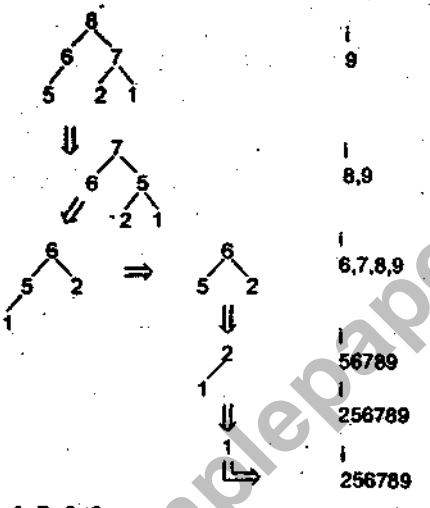
1, 2, 5, 6, 9, 8, 7



Ans. First of all we have to build the max-heap using given data



Now according to algorithm following steps will be followed



The final sorted data is 1, 2, 5, 6, 7, 8, 9.

Q.1. (*b*) Find the solution of the following recurrence relation in O-notation

$$T(n) = 8T\left(\frac{n}{2}\right) + 3n^2$$

where n is an integer power of 2 and greater than 1.

Ans.

$$T(n) = 8T\left(\frac{n}{2}\right) + 3n^2$$

$$n^{\log_2^8} = n^{\log_2^2} \Rightarrow n^3$$

$$f(n) = n^{\log b^9 - \epsilon} \Rightarrow > 0$$

$$T(n) = \theta\,(n^{\log b^9})$$

$$\boxed{T(n) = \theta\,(n^3)}$$

Q.1. (*c*) Develop and analyze an algorithm to determine whether a given $N \times N$ matrix. A has the metric property (that is, for all values of $1 \le i, j, k \le N$, $a_{ij}\ a_{ik} + a_{kj}$) or not.

Ans.

```
metric (A)
    count = 0
For K = 1 to N
  for i = I to N
    for j = J to M
      if a_is > a_ik + a_kj
        count = cot + q
If (count = = 0)
```

return yes
else
return No.
Analysis
$T(n) = O(N^3)$

Q.1. (*d*) Write Master's method for solving recurrence relations of different types.

Ans. Them aster theorem provides a solution for measures at the form

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

where $a \leq 1$, $b > 1$ are constants and $f(n)$ is an asymptotically positive function

We interpret $\frac{n}{b}$ to mean either $\left\lfloor \frac{n}{b} \right\rfloor$ or $\left\lceil \frac{n}{b} \right\rceil$.

Then $T(n)$ can be bounded asymptotically as follows :

1. If $f(x) = 0\ (n^{\log_b 9 - \epsilon})$, for some constant then $T(n) = \theta\ (n^{\log_b 9} \epsilon > 0)$.
2. If $f(n) = \theta\ (n^{\log_b 9})$, then $T(n) = \theta\ (n^{\log_b 9} \log n)$.
3. If $f(n) = \Omega\ (n^{\log_b 9} + \epsilon)$ for some constant $\epsilon > 0$, and if $a f\left(\frac{n}{b}\right) \leq cf(n)$ for some constant $C < 1$ and all sufficiently larger n, then
$$T(n) = \theta\ \{f(n)\}.$$

Q.1. (*e*) Discuss the basic steps in the complete development of an algorithm.

Ans. To design an efficient algorithm we must have to meet same basic requirements.

Q.1. (*f*) Write divide and conquer approach for binary search and find its average case complexity.

Ans. In divide and conquer approach of algorithm design we have to go through three basic steps!

Divide
Conquer
Combine

→ Firstly divide the problems in subproblems.
→ Find out the solution of subproblems
→ and combine those solutions.

For the binary search the precondition is that data must be in sorted order.

Binary sena (A, K)

```
l = b, r = h
while (r! = l)
{
    m = ⌊(l + r)⌋ / 2
    if (A [m] = = K)
        return sneakers
    else if (A [m] < K
            l = m + 1
    else
            r = m
    }
```

The average case complexity is $O(\log_n)$

Q.2. Attempt any *four* parts of the following: (5 × 4 = 20)

(*a*) Write an algorithm for inserting a key into a B-tree in a single pass down the tree.

Ans. BT Insert (T, K)

```
r = root [T] ;
if n [r] = 2t – 1 ;
  S = ALLOCATE-NODE ( ) ;
     root [T] = S ;
     leaf [S] = FALSE ;
     n [S] = 0 ;
     C1 [S] = r ;
     BTS child (S, l, r)
     BT Insert of Non full (S, a)
     else BT Insert Nonfull (r, a) ;
```

In Above algorith two new functions have been called ↓. BTS child for splitting and BT In sort Non full for insertion. We discuss these two as follows.

Q.2. (*b*) Prove that the maximum degree of any node in an *n*-node binomial tree is log *n*.

Ans. By the property of binomial tree we know that for the binomial tree β_n,

- There are 2^n nodes.
- The height of tree is n.
- The root has degree n which is greater than that of any other node ;

The proof of these facts are as :

Binomial tree B_n consists of two copies of B_{n-1} and so Bn has $2^{n-1} + 2^{n-1} = 2.2^{n-1} = 2^n$ nodes.

By the property 3.

maximum degree is n

degree ≤ nodes

$\leq 2^n$

Taking log with base 2

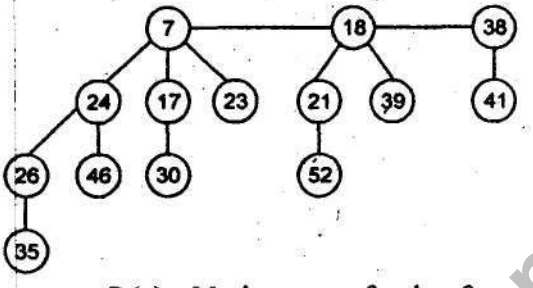
maximum degree = log n.

Q.2. (c) Show that if only the mergeable heap operations are supported, the maximum degree $D(n)$ in an n-node Fibonacci heap is at most [log n].

Ans. Maximum degree Fibonacci heap = $D(n)$

Maximum degree of Fibonacci heap is $D(n)$ then number of node contain = n then Fibonacci heap at most = [log n]

such as



$D(n)$ = Maximum no. of node = 8

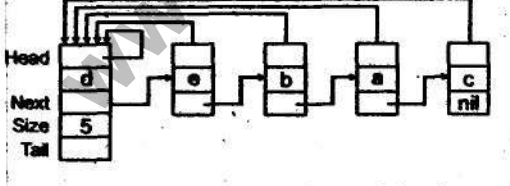
then at most tree is 0 (log n)

= 0 (log 8)

= 3 Hence proved.

Q.2. (d) Write about linked list representation of disjoint sets.

Ans. A simple way to implement a disjoint set data scheme is to represent each set by a linked list. The first object in each linked list serves as set representative such as :



Q.2. (e) Show that A red-black tree with n internal node has height at most 2 log (n + 1).

Ans. x contain at least $2^{bh(x)} - 1$ internal node

x height 0 contain rooted node 0 such at $2^0 - 1 = 0$ each child has at least $2^{bh(x)} - 1$ internal node sub rooted tree contain at least.

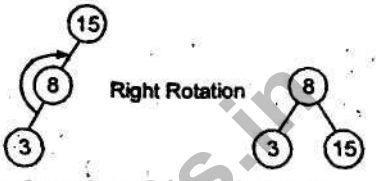
$2^{bh(x)-1} - 1 + 2^{bh(x)-1} - 1 + 1$

$= 2^{bh(x)} - 1.$

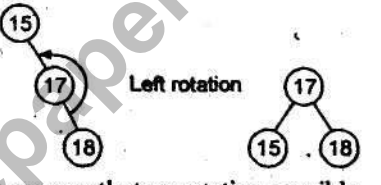
Q.2. (f) Argue that in every n-node binary search tree, there are exactly $n - 1$ possible rotations.

Ans. If n number of node in a tree then $n - 1$ rotation are possible we give the argument with the help of example suppose n = 3 number of node.

15, 8, 3



If n = 3 number of node 15, 17, 18



Here are exactly two rotation possible.

Q.3. Attempt any *two* parts of the following: **(10×2=20)**

(a) Show how Prim's algorithm can be implemented using heap. What would be the time complexity of the algorithm ?

Ans. MSTP (G, W, r)

```
for each u ∈ V [G]
    K [u] = ∞ ;
    P [u] = NIL ;
K [r] = 0 ;
Q = V [G] ;
While Q ≠ Φ
    u = Extract – Again (Q)
    for each V ∈ Adj [u]
        if v.∈ α && w (u, v) < K [u]
            P [v] = u ;
            K [v] = w [u, v]
```

In above algorithm performance depends on how the min-priority queue a is implemented. If α is implemented as a binary min-heap, we can use the build-min-heap procedure to perform the initialization in 0 (V) time. The body of while loop is executed |V| times and since each Extracted-Min operation takes 0 (log V) time, the total time for all calls to Extract-Min is 0 (V log V). The nd for loop is executed 0 (E) times altogether, since the sum of the lengths of all adjoining lists is 2|E|. Within the for

loop, the test for membership in Q can be implemented in constant time by keeping a bit for each vertex that tells whether or not it is in Q and updating the bit when the vertex is removed from Q. The assignment involves an implicit Decrease – Key operation on the min-heap, which can be implemented in binary min-heap, in 0 (log V) time. Thus the total time for prime's algorithm is

$$0 (V \log V + E \log V) = 0 (E \log V)$$

Q.3. (*b*) Give a dynamic programming solution for the subset sum problem. Analyze the complexity of the algorithm.

Ans. Sub Set Sum Problems: Given positive number w_i $1 \le i \le n$ and m this problems calls for finding all subset to the w_i whose sums are m

Ex. if $n = 4$ $(w_1, w_2, w_3, w_4) = (11, 13, 24, 7)$

and $m = 31$, the distorted subset of (11, 18, 7) and (24, 7). Now the two solution are described by the vertex (1, 2, 4) and (3, 4)

Another formulation of subset some problems each solution represent by an n triple $(x_1 x_2 .... x_n)$ such that $x_i \in (0, 1)$ $1 \le i \le n$.

Then $x_i = 0$ if w is not chosen and $x_i = 1$ if w_i is choosen. The solution are instance (11, 0) and (0, 0, 11).

Ex. Given three type of item with following weight and values.

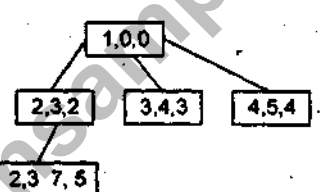
$T = <T_1, T_2, T_3>$

$w = <2, 3, 4>$

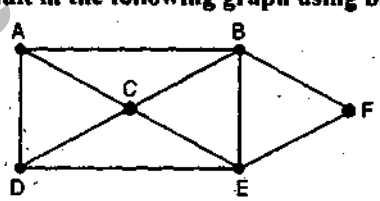
$V = <3, 4, 5>$

w_i V_i w_i

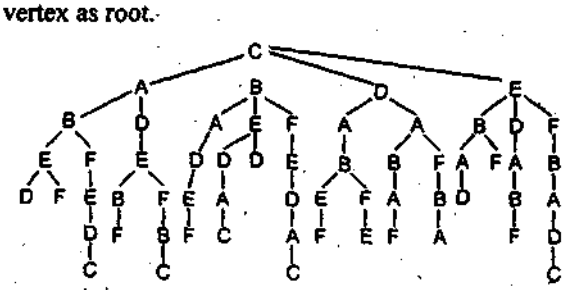
Solution.



Q.3. (*c*) Find the Hamiltonian circuit in the following graph using backtracking.



Ans. Display we make the vertex as root.



All the above path in which (√ √) sign is indicated is solution using as root there are other solution also charging the other vertex as root.

Q.4. Attempt any *two* parts of the following:- **(10×2=20)**

(*a*) (*i*) Prove the correctness of Kruskal's algorithm.

(*ii*) Show how to compute transitive closure of a graph using Flog d-warshell's algorithm for all pairs shortest path.

Ans. MST-Kruskal (G, w)

1. $A = \{\}$
2. For each vertex $u \in v$ [G]
3. Makse-set (u)
4. Sort the edges of E into non decreasing order by weight w.
5. For each edge $(u, v) \in E$, taken in non decreasing order by w.
6. If find set (u) F find set (v)
7. $A = AU \{u, v\}$
8. Union (u, v)
9. Return A.

Any algorithm is said to be create if it satisfies three conditions

Initialization

Maintenance and

Termination.

Above algorithm satisfies these 3 conditions.

Line 1–3 initialize the set A to the empty set and create $|V|$ trees, one containing each voltage. The edges in E are sorted into non-decreasing order by weight in line 4. The for loop in line 5–8 checks for each edge (u, v) whether the end points u and v belongs to different trees. In this case the edge (u, v) is added to A in line 7, and the vertices in the two trees are merged in line 8.

Let G be a weighted graph and let E'CE if E is contained in a MIST T and e is the smallest edge in E–E' which does not create a cycle E'UECT.

Sortest implementation: Sort the m edges in 0 (m log m) time for each edge in order test whether it creates a cycle the forset we have thus for built if so discard', else add the forest with a BFS/DFS this can be done in 0 (n) time.

Kruskal's algorithm builds up connected components any edge where both vertices are in the same connected component create a cycle. Thus if we can maintain which vertices are in which component fast we do not have test for cycles.

If we can test component in 0 (log n) we can find the MST in 0 (m log m)

(*ii*) Transitive-closure (G)

n = |V [G]|

for i = 1 to n

  for j = 1 to n

    if I = j or (i, j) ∈ E [G]

      $t^{(0)}_{ij} = j$;

    else $t^{(0)}_{ij} = 0$;

for k = 1 to n

  for i = j to n

    f = for j = 1 to n

      $t^{k}_{ij} = t^{k-1}_{ij} V_1 (t^{k-1}_{ik} \times t^{k-1}_{kj})$

Return $T^{(n)}$

Q.4. (*b*) Give an algorithm for topological sorting of a directed acylic graph.

Ans. 1. For each vertex $u \in v$
2. do in degree [u] $\leftarrow 0$
3. For each vertex $u \in v$
4. do for each $v \in$ Adj [u]
5. do in degree [v] $\leftarrow$ in degree [v] + 1
6. Q $\leftarrow \phi$
7. For each vertex $u \in v$
8. do if in-degree [u] = 0
9. then ENQUEUE ($\in$, u)
10. while Q $\neq \phi$
11. do $u \leftarrow$ dequeue (Q)
12. OUTPUT u
13. for each $V \in$ Adj. [u]
14. do in-degree $\leftarrow$ in degree [u] – 1
15. if in degree [V] = 0
16. then enqué (Q, v)
17. do if degree [u] = 0
18. then report that there is cycles

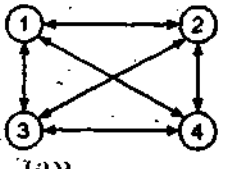
To apply an algorithm on Directed Acylic Graph we need to sort topologically a graph. For example if we have to find out single source shortest plate in directed acylic graph. We have to follow following steps DAG shortest Paths (G, w, S)

1. topologically-sort the vertices of G
2. Initialize-single-source (G, s) || initializing the frequency value to vertices.
3. For each vertex, taken in topologically sorted order.
4. For each vertex u $\in$ Adj [u]
5. Relax (u, v. w) || Relax modify the value to vertices.

So we conclude that we must have to calculate topological sorting. This is done as follows :

Q.4. (c) Give a recursive solution to travelling sales person problem.

Ans.



$$g(i, s) = \min j \in s \{C_{ij} + g(j, s - \{j\})\}$$

$$g(1, \{2, 3, 4\}) = \min (C_{12} + g(2, \{3, 4\})$$
$$C_{13} + g(3, \{2, 4\}$$
$$C_{14} + g(4, \{2, 3\}$$

Q.5. Attempt any *two* parts of the following: **(10×2=20)**

(a) Write Knuth-Morris-Parts algorithm for starting matching.

Ans. KMP (T, P)

n = length [T]
m = length [P]
p = prefix (P)
$q = 0$ for i = j to n
while q > 0 & & P [q + 1] $\neq$ T [i]

```
                q = p (q)]
if p [q + 1] = T [i]
                q = q + 1     if q = m
print "pattern after shift" 1–m ;
                q = p [q] ;
```

In above algorithm we call a function named prefix it is defined as :

```
Prefix [Pa]
                m = length [Pa]
              p [i] = 0 ;
                K = 0 ;      for     q = 2 to m
while K > 0 & & Pa [k + 1] ≠ Pa [q]
                K = p[K]
if        Pa [K + 1] = Pa[q]
                K = K + 1 ;
              p [q] = K ;
return p
```

Q.5. (*b*) Write a randomized algorithm for two-dimensional convex-hull problem.

Ans. We assume that the vertices of a given convex polygon is arranged in counter clock worse order. $0 < V_1 V_2 .... V_n >$ and indexing of vertices is done module n thus $V_0 = V_n$. Thus for a polygon of n sides the vertices might be arranged as $V_{2-1} V_i$. It can be observed that line segment $V_i V_j$ is a chord. If these two are non adjacent sides where $i < j - 1$. Instead of convex polygon of consider the simple polygon then we will extra requirement in which the interior of the segment must be internal of 0.

A polygon can be subdivided by any chord of the form of $< V_i, V_i + ... V_j >$ and $< V_i V_{j+1} .... V_i >$.

Q.5. (*c*) Show that Hamiltonian cycle is in NP class of problems.

Ans. The class NP consists of those problem that are verifiable in polynomial time, means if there is a 'Certificate' of a solution then are can verify that the certificate is correct in time polynomial in size of the input to the problem.

In Hamiltonian–cycle problem, given a directed Graph G = (V, E), a certificate is a sequence $< V_1, V_2, V_3, ... V_{|V|} >$ of $|V|$ vertices. It can be checked in polynomial time that $(V_1, V_{in}) \in E$ for $i = 1, 2, 3, ... |V|_{-1}$.